TES Science Investigator-Led Processing System

Eugene Chu, Denis Tremblay, Kevin Croft, and Andy Griffin

***Abstract*—The Tropospheric Emission Spectrometer (TES) is one of four instruments onboard the Earth Observing System Aura satellite. The TES Science Investigator-led Processing System (SIPS) Facility performs production processing of all TES science data. When the TES project was proposed in 1988, its science algorithms were still evolving, but it was already understood to be at least two orders of magnitude more complex than its NASA predecessor, the Atmospheric Trace Molecule Spectroscopy (ATMOS) instrument. In addition, the expected data volume of the TES instrument would be more than 1000 times greater than that of ATMOS. Development of the TES SIPS faced a number of technical challenges. It also would have been impractical and prohibitively costly to develop the TES SIPS facility without carefully deploying the computing technologies that have recently become available. This paper describes how the challenges were met in the development of the facility by making use of evolving hardware technology and software refinement. The process revealed that the architecture of the hardware implemented was highly dependent upon the processing algorithm, and a stable algorithm was needed early in the hardware development process for performance analysis and benchmarking.**

***Index Terms*—Atmospheric measurements, calibration, computer architecture, computer facilities, data processing, Fourier spectroscopy, Fourier transform, multiprocessing, optical interferometry, parallel processing, processor scheduling, spectroscopy, terrestrial atmosphere.**

I. Introduction

THE TROPOSPHERIC Emission Spectrometer (TES) is a Fourier transform spectrometer (FTS) that measures infrared (IR) emissions from the Earth's surface and atmosphere [1]. Most of the measurements are performed in 29-h periods called global surveys, repeated every two days. During the periods between global surveys, special observation measurements are performed. The total volume of data collected in each global survey is approximately 47 GB. The volume of special observation data is significantly less.

The primary function of the TES Science Investigtor-led Processing System (SIPS) is to run the product generation executives (PGEs) that make up the science processing software to process the interferograms generated by the TES instrument into two types of earth science data type (ESDT) prodcuts. These are the level 1B (L1B) products, which are the interferograms converted into calibrated spectra, and the level 2 (L2) products, which are the vertical profiles of the volume-mixing ratios (VMR) of the chemical constituents in the air columns at each observation scene. These ESDTs are sent to the Distributed Active Archive Center of the Aura Science Data Center at NASA's Langley Research Center (LaRC ASDC DAAC) for archiving and distribution to the user community.

A basic throughput requirement on the SIPS is the ability to process the data collected in one global survey in the two day period between surveys. In addition, the SIPS must also be able to reprocess data due to any processing failures in the first attempt, and as required by expected changes in science algorithms. Combining these processing requirements, the SIPS must provide the capacity to process TES data at a minimum of twice the real-time rate. So, the specified requirement was to provide the capacity to process at three times the real-time rate. These throughput requirements drove the scale of the processing hardware needed in the SIPS. These requirements were derived from a combination of the known computation requirements of previous projects that performed similar types of computations at the time that the TES project was proposed and more recent benchmarks which ran actual TES processing algorithms on available computing hardware. The project that provided the most relevant computational benchmarks was the ATMOS experiment [2].

A. Comparison Between ATMOS and TES

As spaceborne FTS instruments become more sophisticated, their algorithmic and computational needs become more complex. To illustrate the increase in complexity, we present a comparison between TES and the ATMOS experiment. ATMOS was an IR FTS first flown as part of the STS-51B (space shuttle) SpaceLab-3 mission in April/May 1985. Table I summarizes the significant differences between the two experiments in terms of physical attributes and algorithm capabilities.

The TES algorithm has greater complexity due to three main factors: 1) greater number of instrument configurations; 2) more sophisticated theoretical basis and assimilation of non-TES spaceborne information (initial guess); and 3) much larger data volume. The ATMOS retrieval estimates four parameters: continuum level, continuum tilt, frequency shift, and gas VMR. This estimation is repeated for each of the 28 gases at each of the 100 layers. The temperature and pressure profiles were fixed as they were taken from a table. The TES retrieval has several steps. The first step simultaneously estimates 85 atmospheric layers for temperature O_3 and water vapor in addition to surface and cloud properties for a total of 261 parameters. Steps 2) and 3) retrieve CO and CH_4, respectively.

The processing of a TES global survey consists of the following steps.

Manuscript received April 29, 2005; revised September 7, 2005. This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

E. Chu is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: chu@tes-mail.jpl.nasa.gov).

D. Tremblay, K. Croft, and A. Griffin are with Raytheon ITSS, Pasadena, CA 91101 USA (e-mail: dtremblay@tes-mail.jpl.nasa.gov; kcroft@tes-mail.jpl.nasa.gov; agriffin@sdsio.jpl.nasa.gov).

Digital Object Identifier 10.1109/TGRS.2005.863713

TABLE I
ATMOS AND TES ALGORITHM AND INSTRUMENTAL CHARACTERISTICS

	ATMOS	TES
Number of Detectors	1	64
Number of Focal Planes	1	4
Number of Optical Filters	4	12
Resolution (cm^{-1})	0.01	0.015, 0.06
View	Solar Occultation	Nadir, Limb
Retrieval Estimation Methodology	Measure in transmission only, using “onion peeling” retrieval	Simultaneous optimal estimation of all parameters and all layers
Number of Atmospheric Layers	100	85
Number of Retrieved Gases	28	4 (current)
Data Volume	3 GB total	47 GB per Global Survey, 42 TB total (Estimate)

- Level 1A: Interferogram reconstruction and geolocation.
- Level 1B: Conversion of interferograms to spectra and radiometric and frequency calibration of spectra.
- Level 2: Generation of vertical profiles of VMRs.

II. Level 1A Processing

A. Interferogram Reconstruction

The first step in the processing of TES science data is in the Level 1A (L1A) subsystem [3]. The data collected by the TES instrument from each global survey is provided by the LaRC ASDC DAAC as Level 0 (L0) files. These files contain packets generated by the data system onboard the Aura spacecraft which include the interferogram data, instrument performance data, health and status information, as well as spacecraft telemetry information. The L0 data are separated into various data files based on their data types, sequence numbers, scan numbers, and focal planes. A global survey consists of 16 orbits, with each orbit divided into 72 observation sequences, for a total of 1152 sequences. Each sequence consists of seven observations (or scans), two nadir scans, three limb scans, and two calibration scans. Each scan produces an interferogram from each of the 16 detectors on each of four focal planes, for a total of 64 interferograms. The result is 32 256 data files, with each data file containing 16 interferograms, taking approximately 47 GB of storage (there is no reduction between L0 and L1 data). These files are provided to the L1B subsystem for the next stage of processing.

B. Geolocation

The second part of L1A processing is target geolocation [4]. The geolocation algorithms compute nadir footprints and limb ground locations (latitude, longitude, and elevation) of the observed target areas, as well as associated solar and instrument line of sight (LOS), nadir and azimuth angles, and Doppler shift. The algorithms use spacecraft ephemeris to determine its location, and a combination of the spacecraft attitude and instrument pointing control system (PCS) to determine the ground location of each observation.

III. L1B Processing

This subsystem consists of the second most compute intensive tasks of the TES science software. The L1B PGEs convert the observed digitized interferograms into radiometrically calibrated spectra [5].

The first step is to convert the interferograms into frequency spectra using the fast Fourier transform (FFT) algorithm. The FFT is applied to all 516 096 interferograms collected for each global survey.

The next step is to generate the calibration tables. The two calibration scans at each sequence consist of a scan of cold space and a scan of an internal hot black body radiation source. The spectra of the calibration scans collected over the entire global survey are phase aligned, averaged, and saved as a calibration table. This will create 384 calibration table files amounting to approximately 150 MB of data for each global survey. They may be frequency interpolated to match the frequency spacing of the target spectra as needed.

The target spectra phase alignment uses the complex calibration methodology [6]. The cost function to be minimized is the norm of the imaginary part of the radiometric ratio. While the hot black body spectrum is kept as reference, the algorithm applies a phase rotation that corresponds to an optical path difference of an integer number of laser fringes ranging from -50 to $+50$ for both the cold space and target spectra. The algorithm searches through a total of 10 201 possible combinations. To reduce the computational load, the radiometric ratio uses only reduced resolution spectra. This process is repeated for each of the 368 640 target spectra, and accounts for about 40% of the L1B calculations.

The second part of this step computes the noise equivalent spectral noise (NESR) using the Shannon FFT interpolation method [7]. The two-sided interferogram of each radiometric ratio spectra is zero padded to an array size 64 times larger than its original size. After computing the FFT of the expanded interferogram (up to 4.8 million data points each), the NESR is found by linearly interpolating the data to the L2 output grid.

In a final resampling process, another Shannon FFT operation corrects for various frequency scale effects such as Doppler shift. The two Shannon FFT operations account for another 40% of the L1B processing time.

The L1B target processing output files comprise the L1B product binary files, each containing the 64 spectra of the same scan at L2 spectral grid. A global survey will produce 5760 target spectra files amounting to approximately 38 GB of data. These binary files are the input data for the L2 retrieval programs. In addition, the same spectra are written into HDF-EOS5 format files as part of the L1B ESDT that is sent to the ASDC DAAC for archiving and distribution.

IV. L2 Processing

The most computationally intensive part of processing TES data is the L2 retrieval process. The retrieval algorithm [8] infers atmospheric temperature and the concentrations of several

TABLE II
PROCESSING AND FILES AS A FUNCTION OF THE SUBSYSTEM

	Number of Science Files	Number of other Files	Data Volume	Number of Processors	Processing Time
L1A	32256	24192	47 GB	4	8 hr
L1B	6154	16	38 GB	16	48 hr
L2	12	1	5 GB	60	48 hr

atmospheric constituents from the observed radiance. The observed radiance is a function of the emitted surface radiance, thermal emissions from the atmosphere, and absorption of the emitted radiance by atmospheric constituents.

The retrieval process starts with the generation of a forward model of the radiance based on the initial guesses of atmospheric and surface states of the target area using climatology information at the time of the observation. The troposphere is modeled in 85 discrete atmospheric layers. The radiance at the top of the troposphere is computed by calculating the thermal emissions (a function of temperature and frequency) and the absorptions (a function of pressure and temperature, and also dependent on atmospheric species) of each chemical constituent at each atmospheric layer. The radiance contributions are calculated from one layer to the next using radiative transfer methods. The radiative absorption coefficients (ABSCO) of each chemical species are derived from the HITRAN spectroscopic parameter compilation [9]. The ATMOS retrieval software generated these coefficients during processing. The TES algorithm use a faster method to generate these coefficients; the coefficients are pregenerated for various chemical species at prescribed temperature, pressure, and frequency ranges, and stored in files. There are currently approximately 620 GB of these ABSCO data stored in about 55 000 files. Their values are interpolated to the frequencies, temperatures, and pressures of the forward model at runtime.

The forward model algorithm uses analytic computations of the Jacobians (i.e., partial derivatives) of the radiance with respect to the estimated atmospheric parameters. The radiance computed from the forward model is then compared with the observed radiance using a nonlinear least-squares solver. The differences between observed and modeled radiances are used with the Jacobians to adjust the initial guesses to the next iteration of the forward model. These steps are repeated until specified convergence criteria are met.

Table II shows a summary of current processing resources required and files and sizes generated by each subsystem.

V. PROCESSING IMPLEMENTATION

The benchmarks of retrieval runs in processing ATMOS data provided a baseline performance metric using existing (circa 1990) computing technology. However, applying that benchmark to processing TES data to estimate the required computation hardware yielded a system that was impractically large and prohibitively expensive for that time. For example, using the compute platform that ATMOS implemented, a MIPSCO M/2000, a 25-MHz reduced instruction set computer (RISC) workstation with an estimated performance rating of approximately 22 Spec92FP, would have required about 12 000 units to meet the computing requirements just for running the TES L2 retrieval. In addition, using the existing disk drives, CDC-9720-1.2G, a 9-in form factor disk drive with approximately 1-GB capacity, would have required 200 units to support the minimal storage needs of processing a single global survey, but the data processing industry was expected to advance the processing capabilities of computers and increase the density of storage devices. At that time, computer manufacturers generally met expectations in the rate of advancement of the performance of their processors. So, the TES project planned that, by the time that the production facility was expected to be implemented (approximately 2002), the available compute platforms would provide sufficient performance in a small enough foot print to start making this practical and still fit into the available budget.

In addition to depending on faster computer hardware, a nontrivial amount of work was required to implement the science algorithms into production software. Most of the software development efforts centered around the L2 subsystems. The details of the science software development tasks are covered in another paper published in [10]. During the development process, the hardware and software engineers worked closely to ensure that each group understood the needs and limitations of both the hardware and software to make sure development on each side could accommodate the other.

The first program used by TES for performing L2 retrieval tests on simulated data was called TWPR, an acronym for TES working prototype retrieval. This program was made up of several software algorithm components, developed by various scientists and engineers who worked on the TES project, each of which performed specific functions within the overall retrieval process. From the beginning of the TES project in 1996, a working prototype was needed as quickly as possible to support science algorithm analysis. As a consequence, the TWPR software was not designed with optimization in mind. Furthermore, the interfaces between the algorithm components were not well defined. The performance of TWPR on computing hardware that was state of the art at the time was not sufficient to practically meet even the most basic throughput requirements. For example, based on run times obtained in 1998 using a Sun Enterprise 4500 server with two 336-MHz processors, L2 retrievals on a global survey would have required over 250 000 h to complete. This was an estimate based on running small numbers of retrievals using simulated data. Due to the extremely long run times, no simulated full global surveys were ever actually processed up to this time. Assuming processing throughput scaled linearly with the size of the compute system, using this computing technology would have required approximately 1000 fully configured Enterprise 4500 systems in order to provide sufficient computing throughput to keep up with the rate at which global surveys would be acquired. More testing revealed that the performance of the L2 retrievals did not scale linearly with the size of the system it was run on. A system containing many processors became geometrically less efficient as more retrieval jobs were run concurrently on it. The problem was that the multiple jobs that were initiated on each system to make use of its multiple processors would all contend for the same resources: the ABSCO files and the data bus that the data have to travel across. The tests showed that the system configuration that provided the most effective use of system resources with minimal contention between jobs was one containing two processors. This indicated that a different hardware architecture was needed; instead of a large system with many processors, each system would contain just two processors, and many such systems (or nodes) would be deployed in parallel.

A concerted effort was started in 1999 to help make the TWPR program run faster. A small team consisting of several software engineers and science algorithm developers carefully analyzed the program to determine where it spent most of its time. They determined which aspects of the calculations could be reduced, streamlined, or eliminated, and still be able to generate valid science results. Many parts of the software were then rewritten to operate more efficiently.

The first major bottleneck was found to be in the way the ABSCO files were accessed and processed. The program spent most of its time waiting for the ABSCO data to be read from disk. In addition, when retrieving a particular species, only a very small fraction of the observed spectra, and, thus, the associated ABSCO data, contained significant information. One of the first modifications to the TWPR program was to make it process only the small windows around the parts of the spectra that contained the absorption lines. These "microwindows" significantly reduced the amount of spectral data used in the forward model algorithm, thereby reducing the amount of frequency-dependent calculations by an order of magnitude.

The calculations were further reduced by a factor of two by using half the spectral resolution of the data. This also meant that the ABSCO data could be stored in half its original spectral resolution, which reduced the storage requirements of the ABSCO tables to 310 GB from 620 GB. Further speedup was achieved by changing the program to read only those parts of the ABSCO files that contained the microwindow data.

Other software refinements included combining functions that were executing independently and using common information to eliminate processing steps that were no longer needed. The combined efforts achieved a reduction in run times of approximately two orders of magnitude, while the retrieval results retained their original criteria for quality. The new program, the first version of the production code, was released in 2001, called Earth Limb And Nadir Operational Retrieval (ELANOR).

Further development on ELANOR showed that the retrievals still spent a significant amount of time accessing the ABSCO files. In a single retrieval, accessing the ABSCO files can account for up to 20% of the total run time, but its CPU time was typically less than 6% of the total CPU time. Initially, the ABSCO files were stored on a single high performance RAID array, and served out to various users by its host. The file access times increased exponentially as multiple retrievals were being run simultaneously. Additional development efforts concentrated on further reducing the access times for the ABSCO files. One approach was to simply replicate the ABSCO files on another RAID server. The final solution was to install sufficient storage on each processing node to hold its own copy of the ABSCO files. This was made more practical with the decreasing costs of disk storage systems.

Another change was made to the way the ELANOR program accessed the ABSCO files. Since the retrievals processed spectral microwindows, they did not need the full ABSCO tables. A microwindow caching scheme was developed to save the parts of the ABSCO files that were used in retrievals into smaller files. These smaller files required less time to access than the full ABSCO files, which further reduced run times. The standard retrievals were run in five iterations, and all iterations after the first would benefit from the microwindow cache. This enhancement reduced total ABSCO access times by another factor of two.

The L1B prototype software was developed using the Interactive Data Language (IDL) provided by the Research Systems, Inc. (RSI). This is an interpretive language originally intended for users to quickly develop and test modeling and simulation algorithms. This kind of programming model would not normally fit well within a production environment where high speed and efficiency are important. However, IDL has extensions that allow it to integrate user programs written in compiled languages such as C, where most of the computations are performed. The L1B production software makes use of this software configuration to perform most of its compute intensive tasks. It uses an IDL based front end for implementing the high level L1B algorithms and for managing data inputs and outputs, and a C program which uses compiled libraries, such as FFTW, to perform most of its low level numerical computations. The entire L1B code could have been translated into a compiled language such as C or C++ at an estimated cost of about two man years. However, there was no significant performance enhancement expected from this conversion, because of the way the L1B code used IDL and C programs. The L1B processing currently represents about 25% of the total processing that needed to be performed on a global survey.

VI. Hardware

At the same time that the science software was being refined, the SIPS hardware development team was studying compute hardware produced by various manufacturers to find a candidate that provided the best performance at the lowest cost. A stand-alone version of the ELANOR program was created, and a retrieval test using a typical set of parameters was used for performing benchmarks. The program was ported to the various platforms using their native compilers, and their run times were recorded. Their results were also compared to those of the reference system, the Sun server on which the program was originally developed, for accuracy.

It was also determined that, in addition to raw computing capabilities and price, a critical requirement of a system was to have sufficient support from the commercial software companies that provided tools and utilities used on the original development platform. The Sun SPARC platform enjoyed support from a broad range of third party software market, which made it more popular to users. Other platforms considered by the TES project needed similar levels of support for software such as compilers, databases, configuration management tools, and other utilities, such as IDL.

The hardware study found a number of low cost commodity microprocessor-based computing systems that were very good candidates for the TES SIPS. The performance of microprocessors produced by Intel and Advanced Micro Devices (AMD), known as the x86 CPUs, were surpassing those of the big server manufacturers such as Sun and HP. At the same time, their costs were typically much lower than the big servers. In general, a complete small server using an x86 microprocessor would cost about one tenth as much as a server of similar performance from one of the big manufacturers. These low-cost servers ran the Linux operating system, an open source software package with a vast support network in the free software community. Most importantly, various software vendors were porting their products to the Linux operating system, so the same functionalities of these software products that were available on the large expensive servers were becoming available on the less costly platforms. These low cost x86 platforms made the development of a large computing cluster much more economically feasible; it was now possible to purchase sufficient computing power to meet our throughput requirements and our budget, and still be able to support a working environment that the developers and science team were accustomed to.

The new x86 based computing platform required further development efforts in the software subsystems to port the software from the Sun Solaris environment. A side effect of these efforts was to reveal many errors in the software that were hidden by the Solaris operating system and its compilers. So, the move to the new platform improved the portability and quality of the software as well as enabling them to run faster. One issue that arose was the data formats used on these two platforms; the Sun Solaris running on SPARC hardware used Big-Endian data representation, while the x86 platform used Little-Endian. This required careful handling of data formats on the different platforms used by the TES project. Once the format differences were understood, simple coding in the PGEs running on different platforms correctly dealt with their different data types.

With each new generation of hardware that became available, their performance usually improved as well. For example, at the time that the SIPS hardware was being procured, the microprocessor that was found to have the lowest price to performance ratio running the TES L2 retrieval software was the AMD Opteron. This processor has a 64-bit architecture but is fully compatible with the 32-bit x86 architecture and Linux environment that was used to develop the TES science and production software. The initial systems used by TES would be running in 32-bit mode, but these processors allowed future migrations into 64-bit mode, where it can accommodate working with larger data sets if the algorithms should grow in that direction. The available disk storage at the time provided 250 GB of capacity on a low profile drive. So, it was possible to configure a physically small server that occupied one rack unit of space (1.75 in of rack height), but contained two very fast processors, 4 GB of memory, and 500 GB of disk space, which was sufficient to store the entire half resolution version of ABSCO tables and still leave enough space for the SIPS and PGE programs to work with. This configuration is the basic compute node used in the SIPS cluster. The SIPS hardware was sized to meet the throughput requirements of three times real time rate. Fig. 1 shows a block diagram of the major hardware components of the TES SIPS.

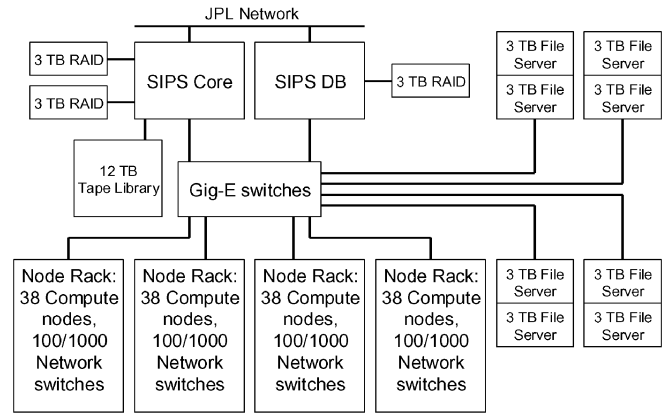


Fig. 1. TES SIPS hardware diagram.

A. Compute Nodes

There are 152 compute nodes in the first phase of the compute cluster installation. There are 38 nodes mounted in each rack. Each rack also contains two managed power strips, two 1000/100/T Ethernet switches, and a terminal server for connection to the serial console ports of each node. Each compute node in the cluster consists of a dual processor server using two AMD Opteron 242 CPUs, 2 GB of RAM, two IDE disk drives with 250-GB capacity each, and two Ethernet ports. Each Ethernet port is connected to a separate switch in the rack. Each switch uplinks to the central (core) network switches through Gig-E ports. The managed power strips allow each node to be remotely powered on or off through a network interface.

B. File Servers

There are eight file servers in the first phase of the SIPS cluster installation. These file servers are used for staging the data used as inputs to the PGEs, and to store their outputs. Each file server consists of a dual processor server using two AMD Opteron 244 CPUs, 4 GB of RAM, two IDE disk drives with 250-GB capacity each, 2-GB Ethernet (Gig-E) ports, and a 3-TB disk array connected through an Ultra160 SCSI interface. Each Gig-E port is connected directly to a core network switch.

C. SIPS Core Server

The SIPS core server is hosted on a Sun Enterprise v480 server. This was necessary because the SIPS core software was developed on a Sun/Solaris environment. The software performs little computation, so processor throughput was not an issue. However, it was necessary for the system to interface to various types of hardware, including a LTO library for data archiving, and move all the TES data between them. These interfaces have been well worked out in the Solaris environment. In this system, the cost of porting the code to Linux would have been at least another man year, and would not have gained any increases in performance or functionality.

This server consists of four UltraSPARC-III/1.05-GHz CPUs and 16 GB of RAM. The server uses two disk arrays of 3 TB each, attached through an Ultra160 SCSI interface. The server also uses 2 Gig-E connections to the core network switches as well as a separate Gig-E connection to the JPL public network. The core server also uses a large tape robot library as its local data archive.

D. SIPS Database Server

The SIPS database server is also hosted on a Sun Enterprise 480 server, an exact copy of the SIPS core server. This server can serve as a backup to the SIPS core at reduced throughput rates for both if the SIPS should suffer a major hardware failure. The data base server could also have been hosted on an Opteron server running 64-bit Linux for better price/performance. But this data base needed the larger addressing capability of the 64-bit version of the Oracle data base server for Linux, which was not ready at the time that the system was being implemented. This server uses a 3-TB disk array attached through an Ultra160 SCSI interface. The server is also connected to the core network switches through two Gig-E ports.

E. SIPS Network

The core network switches connect all the major components of the SIPS hardware together. All the servers are connected directly to these switches using Gig-E. The switches in the node racks also connect to the core switches using Gig-E. The core switches are Foundry Network's EdgeIron model 24G, each with 24 Gig-E ports. Each switch has sufficient bandwidth to provide wire-speed communication between any two ports, with all port pairs running concurrently.

VII. Lessons Learned

It is important to identify a version of the working software prototype that provides all the basic science algorithms and functions that need to be performed as early as possible to serve as the baseline for generating the production software. This also allows accurate performance analysis of the software on available hardware platforms as early as possible. An important part of performance analysis is to understand the algorithm sufficiently to determine where to apply the most effective optimization efforts.

A compact, portable, and accurate representation of the most compute intensive part of the software was very important for benchmarking the different compute platforms. This allowed the software to be quickly ported to each platform, thus identifying the hardware that can provide the highest computational throughput for the software that will become part of the production system.

Performance analysis and optimization continues for as long as the algorithm and software changes to adapt to better understanding of the data. This is important to make sure that any changes that significantly affect the production throughput can be quickly identified and mitigated.

The same hardware can offer varying levels of performance in executing the same program depending on how the hardware and the program are configured. The best performance can be achieved from the hardware if its use was tailored to the functions that the program was performing. While the performance of processors advanced almost at the rates that industry predicted, the performance of the peripherals that support processing, especially hard disk drives, did not keep pace. This required special handling of data stored on hard disks to minimize the effects of their latencies.

The throughput performance of a hardware platform is not the only factor in its selection. An important factor for the TES project was how well a platform will be supported by the commercial software vendors and the computing industry in general. When the software benchmarking process started, the x86 platform was not the fastest; the DEC Alpha provided the highest performance of all the platforms tested A decision was made to use the x86 platform as it appeared to have the most potential for performance improvement and future industry support. However, it is a good idea not to get locked into the products of any particular vendor. As an example, the x86 computing platform is commodity hardware that can be purchased from many sources. However, some of the tools that TES used for software development are proprietary, and limited the system configurations that could be used. The TES project was able to work within these limitations created by what is known as "vendor lock in," and it should be avoided if possible.

VIII. Conclusion

The TES science software is very complex and requires tremendous computing resources to execute. The available computing platforms in 1988 would have made processing TES global surveys very costly and impractical. A combination of software improvements and hardware evolution during the development of the TES project made implementing the SIPS computing system feasible.

Early prototypes of the science software will undergo drastic changes before it is developed into production software. Detailed characterization of the performance of the production software was critical in identifying where the bottlenecks were. This enabled more effective application of optimization efforts. These efforts included refining the science algorithms to eliminate unnecessary calculations, streamlining the software implementing these algorithms, and determining the best application of improving hardware technologies.

Major increases in software throughput were achieved by improving the way the ABSCO data were accessed. The first improvement was to reduce the amount of data that needed to be accessed by implementing the microwindows around absorption lines, which also reduced the amount of calculations that were needed. The second major improvement came with the availability of the small but high capacity and high speed disk drives that could be installed onto each compute node. This allowed each node to have their own copies of the ABSCO files, reducing contention for the data. The next improvement was caching the microwindows in smaller disk files. A final improvement was not explicitly implemented, but realized with the large memory installed into each node. This allowed the operating system to cache frequently used data in memory, which effectively eliminated disk access times.

ACKNOWLEDGMENT

The authors would like to thank the following people for their technical assistance in the preparation of this paper: R. Beer, TES project principal investigator; M. Gunson, TES project deputy principal investigator; S. Paradise, TES ground data systems software element manager; R. Toaz, TES ground data systems manager; and J. Worden, TES science forward model lead

REFERENCES

[1] R. Beer, T. Glavich, and D. Rider, "Tropospheric Emission Spectrometer for the Earth Observing System's Aura satellite," *Appl. Opt.*, vol. 40, pp. 2356–2367, 2001.

[2] C. B. Farmer and R. H. Norton, *A High-Resolution Atlas of the Infrared Spectrum of the Sun and Earth Atmosphere from Space*. Washington, DC: NASA, 1989. NASA Ref. Pub. 1224.

[3] H. Worden, R. Beer, K. Bowman, B. Fisher, M. Luo, E. Sarkissian, D. Tremblay, and J. Zong, "TES level 1 algorithms: Interferogram processing, geolocation, radiometric, and spectral calibration," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 5, pp. 1288–1296, May 2006.

[4] K. W. Bowman, E. Sarkissian, D. Tremblay, H. Worden, and J. Zong, "Tropospheric Emission Spectrometer (TES) Level 1 Algorithm Theoretical Basis Document (ATBD), V 2.0," JPL, Pasadena, CA, JPL Int. Doc. D-16479, Aug. 2004.

[5] H. M. Worden and K. W. Bowman, "Tropospheric Emission Spectrometer (TES) level 1B algorithm theoretical basis document (ATBD), V. 1.1," NASA Goddard Space Flight Center, Greenbelt, MD, JPL Int. Doc. D-16479, Oct. 1999. [Online]. Available: http://eospso.gsfc.nasa.gov/atbd/testables.html.

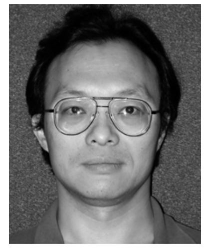
[6] H. E. Revercomb, H. Buijs, H. B. Howell, D. D. Laporte, W. L. Smith, and L. A. Sromovsky, "Radiometric calibration of Fourier transform spectrometers: Solution to a problem with the High-Resolution Interferometer Sounder," *Appl. Opt.*, vol. 27, pp. 3210–3218, 1988.

[7] R. J. Marks, *Introduction to Shannon Sampling and Interpolation Theory*. New York: Springer-Verlag, 1991.

[8] R. Beer, K. Bowman, P. Brown, S. Clough, D. Jacob, J. Logan, M. Luo, F. Murcray, D. Rider, C. Rinsland, C. Rodgers, S. Sander, E. Ustinov, and H. Worden, TES Level 2 Algorithm Thoeretical Basis Document (ATBD) V. 1.1, JPL, Pasadena, CA, Oct. 1999. [Online]. Available: http://tes.jpl.nasa.gov/docsLinks/publications.cfm#docs.

[9] L. S. Rothman, A. Barbe, D. C. Benner, L. R. Brown, C. Camy-Peyret, M. R. Carleer, K. Chance, C. Clerbaux, V. Dana, V. M. Devi, A. Fayt, J.-M. Flaud, R. R. Gamache, A. Goldman, D. Jacquemart, K. W. Jucks, W. J. Lafferty, J.-Y. Mandin, S. T. Massie, V. Nemtchinov, D. A. Newnham, A. Perrin, C. P. Rinsland, J. Schroeder, K. M. Smith, M. A. H. Smith, K. Tang, R. A. Toth, J. V. Auwera, P. Varanasi, and K. Yoshino, "The HITRAN molecular spectroscopic database: Edition of 2000 including updates through 2001," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 82, no. 1–4, pp. 5–44, 2003.

[10] S. Paradise, S. Akopian, K. Croft, K. Fry, S. Gluck, D. Ho, M. Lampel, J. McDuffie, R. Monarrez, H. Nair, S. Poosti, D. Shepard, I. Strickland, D. Tremblay, H. Yun, and J. Zong, "TES ground data system software," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 5, pp. 1343–1351, May 2006.



Eugene Chu received the M.S. degree in electrical engineering from the California Institute of Technology (Caltech), Pasadena, in 1983.

He manages the development and operations of the Science Computing Facility and the Science Investigator-led Processing System for the TES project at the Jet Propulsion Laboratory, Caltech.

Denis Tremblay received the Ph.D. degree in aerospace engineering from the University of Colorado, Boulder, in 1995.

He has been a member of the TES Ground Data System, Raytheon, Pasadena, CA, since 1998. He contributed to the software developement with emphasis on calibration (L1B), geolocation, simulation, integration, testing, and science validation.

Kevin Croft received the degree in geomatics engineering from the University of Calgary, Calgary, AB, Canada.

He was a major contributor to the L2 software optimization effort and developed an end-to-end testing infrastructure for the TES data processing system at Raytheon, Pasadena, CA.

Andy Griffin, photograph and biography not available at the time of publication.